

NAME (from your UF ID): _____ UF ID#: _____
(Please **PRINT**)

----- CEN 6070 Software Testing & Verification -----

Exam 1 -- Summer 2016

You have 90 minutes to work on this exam. It is a "closed-book/closed-notes" test. Pay attention to point values, since you may not have time to work all 20 problems. PRINT your name and UF ID# above NOW and sign the pledge at the bottom of the last page, if appropriate, when you are finished.

PLEASE PRINT – **do NOT write *cursively*** – ANSWERS IN THE SPACE PROVIDED ONLY – **NOT IN THE MARGINS** – PREFERABLY USING A BALLPOINT PEN (electronic or otherwise!) TO INCREASE LEGIBILITY. Good luck!

1. In the "Introduction to V&V Techniques and Principles" lecture, a discussion of alternative definitions of "testing" is presented.
 - a. (2 pts.) Who is the author of the required course reading that most closely reflects this discussion and the conclusions drawn from it?
 - b. (3 pts.) How, specifically, does the author characterize the *importance* of this discussion relative to other things one can learn about program testing?
 - c. (5 pts.) What is the author's preferred definition of testing and what assumption regarding human nature is the premise for his opinion in this regard?
 - d. (4 pts.) Two testing-related terms that directly relate to the discussion of alternative definitions of testing are "validation testing" and "defect testing". Describe what these terms refer to, making the relevant distinction between them clear.

2. (3 pts.) A distinction is sometimes drawn between the terms "Test-Driven" and "Test-First" Development, whereby "Test-First" is taken to be *stronger* than "Test-Driven". What, specifically, is implied by the term "Test-First" that sometimes goes beyond what is implied by "Test-Driven"? Be specific.

3. Two fundamental implications of the example given in class about how long it would take to exhaustively test a simple module with 2 input parameters are: (1) that one cannot, in general, test a program to guarantee that it is error free, and (2) that a fundamental consideration in program testing is (therefore) one of *economics*. Microsoft's Ken Johnston builds on these implications to support a new model of testing in his Guest Lecture, "The Future of Testing is EaaSy."
 - a. (3 pts.) Briefly identify the model by name and describe the basic premise upon which it builds.

 - b. (6 pts.) Describe (i) what "feature flagging" is and (ii) clearly explain how this idea directly supports the application of this model at Microsoft.

4. (3 pts.) Joseph Cutrono used the term "journey test" when comparing and contrasting unit testing, acceptance tests, and UI tests in his Guest Lecture. Briefly describe what he meant by "journey test".

5. (9 pts.) Define each of the following software testing-related terms.

a. alpha testing:

b. integration testing:

c. regression testing:

6. (4 pts.) Reference was made in class and in the lecture notes to a Feb. 3, 2010 Bloomberg article about Toyota having ruled out electronics as a cause of sudden acceleration that had resulted in recalls of millions of its cars and trucks. Which one of the following points was this article used to illustrate? (Choose ONE only.)

a. Test cases must be written for INVALID and UNEXPECTED, as well as valid and expected, input conditions.

b. The owners of cars with electronic throttle systems have yet to fully embrace testing models that call for developers to "become comfortable with testing less and knowingly shipping buggier software faster than ever before."

c. The term "exhaustive testing" is often used rather loosely.

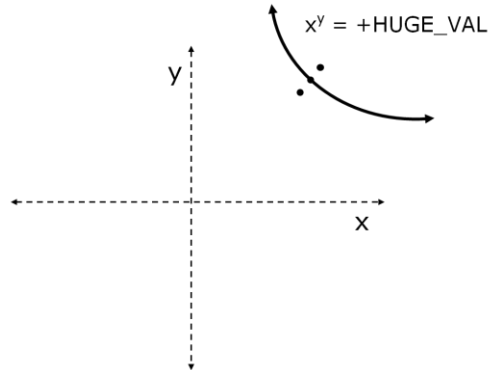
d. In general, the probability of the existence of more errors in a section of a program is directly related to the number of errors already found in that section.

e. Programmers should in principle avoid testing their own programs.

f. It is important to test system versions over a significant period of time to discover latent errors or performance problems (due to memory leaks, buffer/file overflow, etc.)

g. (None of the above)

7. (8 pts.) The graph below was used in class to illustrate the application of a specific test case design technique. Briefly explain its context by identifying (i) what specific technique is being illustrated (e.g., input space partitioning, cause effect analysis, logic coverage, etc.), (ii) the problem it is being applied to (e.g., fmod, sort, exp, etc.), and (iii) what the curve AND points *represent* in the problem. Be specific.



8. Suppose a function that returns 3 Boolean values has 4 inputs, one of which is comprised of 3 disjoint attribute classes, while the others are comprised of 2 disjoint attribute classes each. There are no infeasible input combinations.
- a. (6 pts.) What is the minimum number of test cases needed to achieve "Strong Equivalence Class Testing"? What is the minimum number needed to achieve "Weak Equivalence Class Testing"?
- minimum # of cases needed for *strong equivalence class testing*: _____
 minimum # of cases needed for *weak equivalence class testing*: _____
- b. (4 pts.) Joseph Cutrono, in his recent Guest Lecture, described another combinatorial testing criterion called "all-pairs." What is the minimum number of test cases needed to achieve "All-Pairs Equivalence Class Testing"?
- minimum # of cases needed for *all pairs equivalence class testing*: _____
- c. (3 pts.) Suppose a given implementation of this function has a total of 9 execution paths, 7 of which are feasible. If the function's input space is partitioned solely on the basis of *specified output*, what is the minimum number of test cases needed to ensure coverage of the partitions?

minimum # of cases needed to cover the partitions: _____

9. (20 pts.) Consider the following coverage criteria related to Cause-Effect Analysis:

- (1) AFCCV (All Feasible Combinations of **Cause** Values)
- (2) AE (All Effects)
- (3) AFCEV (All Feasible Combinations of **Effect** Values)
- (4) Strategy #3
- (5) Strategy #3 plus culling rules

For each individual criterion listed, identify (using the labels given above) ALL (and ONLY) the *other* criteria **subsumed** by it. (Note: criterion A **subsumes** criterion B if and only if B will always be satisfied whenever A is satisfied. Informally, this is often written " $A \Rightarrow B$ ".)

For example if you include the label "(3)" in your list for "AE" below, you would be (incorrectly) asserting that " $AE \Rightarrow AFCEV$ ". You need NOT include criteria in their own list (e.g., no need to include "(4)" in your list for "Strategy #3").

AFCCV: _____

AE: _____

AFCEV: _____

Strategy #3: _____

Strategy #3 plus culling rules: _____

10. (6 pts.) Consider the pseudocode program:

```
(1) input (A,B);
    if (A OR B) then
(2)   s1
(3) end_if
```

for Boolean variables A, B. PROVE, using one or more counter-examples, as necessary, that

branch coverage does NOT subsume condition coverage.

Explicitly identify any counter-example(s) used in your proof by identifying the Boolean input values assumed for A and B, the path(s) sensitized, etc. Your counter-example(s) and explanation must make your conclusion clear.

11. A variation on "Test Case Selection Strategy #3" that we considered involves the use of particular "culling rules".
 - a. (5 pts.) Describe the mechanics and combinatorial impact of applying this variation when encountering a 3rd-degree OR-node that must evaluate to True in the process of identifying Cause combinations resulting in a given Effect.
 - b. (5 pts.) Describe the combinatorial savings in general mathematical terms AND briefly explain the rationale for the particular culling approach used.

12. (4 pts.) What potential problem with using techniques such as "Error Seeding" to estimate the number of errors remaining in a program during testing was illustrated in class by way of a pseudocode Quick Sort program? *How* did the program illustrate the problem? Be specific.

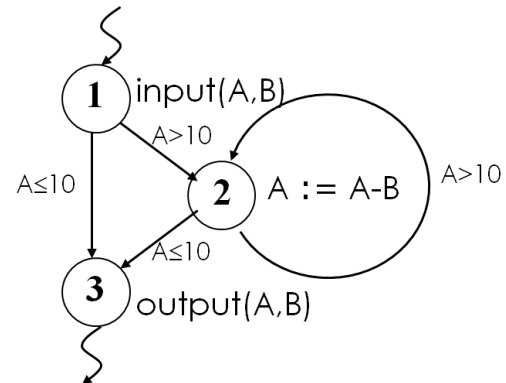
13. (7 pts.) In "*The six essentials of software testing*," Edward Kit describes "the essentials of the software testing process that serve as the foundation" for his book, Software Testing in the Real World: Improving the Process.
 - a. Essential 2: Prevent defect migration by using early life-cycle testing techniques. What specific example does he give of a proven testing technique that can be applied early in the development process?
 - b. Essential 3: The time for software testing tools is now. What **two** "essential tools" does he give to illustrate his point?

14. (13 pts.) Consider the pseudocode program and corresponding control graph below.

```

1. input(A,B)
   while (A>10) do
2.   A := A-B
   end_while
3. output(A,B)

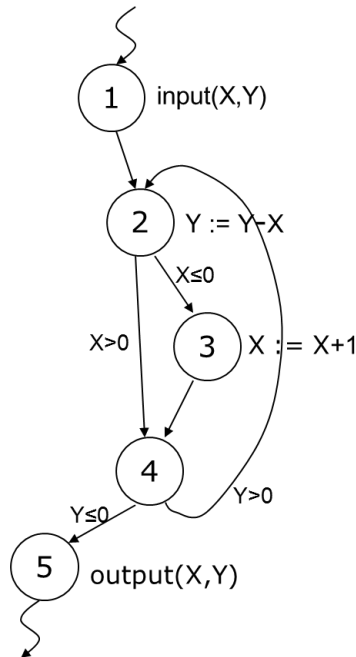
```



In the table below, list all definition-use pairs for variables A and B, and identify all (and only) the associated du-paths for each pair.

[illegible]

15. (7 pts.) Consider the program control flow graph below.



- a. Give the complete *path condition* for path $\langle 1, 2, 3, 4, 2, 4, 2, 4, 5 \rangle$ in terms of the **input values** (X_0 and Y_0) of variables X and Y . (Do not combine or otherwise simplify predicates.)
 - b. Give actual values for X_0 and Y_0 , if any, that would satisfy the path condition for the path given in part (a). If none, write "none".
16. (4 pts.) In their paper, *The Effectiveness of Software Development Technical Reviews: A Behaviorally Motivated Program of Research*, Sauer, et al. apply the behavioral theory of group performance to explain the outcomes of software reviews. Which one of the following, according to the authors, is the "most salient (i.e., important) finding" of the empirical research in which the theory is based? (Circle ONE only.)
- a. That the performance advantage of an interacting group over a nominal group is a function of the level of false positives discovered by individuals.
 - b. That group performance is dominated by the available task expertise.
 - c. That group performance is a positive function of task training.
 - d. That synergy is the primary reason why groups perform better than individuals.
 - e. That decision schemes (plurality effects) dominate interacting group performance.

17. (14 pts.) Match each description below to the **SINGLE MOST APPROPRIATE TERM** related to testing object-oriented software among the following. (Note: terms may apply to none, one, or more than one description.)

- | | |
|-----------------------------|---|
| A. encapsulation | G. polymorphism |
| B. inheritance | H. "observability" interfaces |
| C. unit level O-O testing | K. state machine models |
| D. object classes | L. methods |
| E. use/include relations | M. UML sequence or collaboration diagrams |
| F. higher level O-O testing | N. inspection operations |

- ___ Normally begins with the integration of object classes to form inter-object class functional entities
- ___ Allows variable types and method bindings to change dynamically, thus requiring testers to be aware of the bindings that may occur
- ___ Object-oriented design principle that may necessitate the use of special instrumentation to interpret the effects of methods
- ___ Facilitates the application of partitioning and combinatorial test case design techniques on a state-by-state basis
- ___ Allows object classes to be *specialized* or *extended* from one or more other object classes
- ___ Often focuses on object classes as opposed to individual methods, as testing methods in isolation is not always practical
- ___ Sometimes required by design rules to circumvent the problem of object state not being directly accessible, but results in some overhead

18. (3 pts.) During the discussion of testing tools, reference was made to the wonder and beauty of springtime in Paris...



Which ONE of the following testing tool types was this reference used to clandestinely introduce? (Circle ONE only.)

- a. Boundary value, equivalence class, and special value generators
- b. Statistical test generators (for reliability, availability, and performance testing)
- c. Test Harness Generators
- d. Code Instrumentation tools
- e. Data / Output Comparators
- f. Keystroke Recorder and Playback Tools
- g. (none of the above)

19. (12 pts.) In their chapter, *"Making Meetings Work for Everyone,"* Gause and Weinberg offer several observations and recommendations related to making meetings more productive. For each of the following, circle "is" if the observation or recommendation is included in the chapter, and "is not" otherwise. To compensate for random guessing, your score in points will be 2 times the number of [correct minus incorrect] answers, or 0 – whichever is greater. Therefore, if you are not more than 50% sure of your answer, consider skipping the item.

- | | | |
|---|----|--------|
| a. People who inappropriately come to meetings are likely to start participating to "prove" they were right to be there in the first place. | is | is not |
| b. If the same people are to attend different meetings one right after another in the same room, have the participants drink a strong cup of coffee before starting the second meeting. | is | is not |
| c. Meeting participants should agree in advance that anyone can call one-minute or five-minute time-outs at any point without explaining why. | is | is not |
| d. Noticing what's wrong with your meetings will help diagnose what's wrong with the entire process, and indicate what to prescribe for the malady. | is | is not |
| e. It is important to never allow agreed-upon rules to be changed during a meeting since this could be used as a political tool by one side against another. | is | is not |
| f. One way to see if there is unfinished business at the end of a meeting is to ask each person, "Has every one of your ideas been handled to your satisfaction?" | is | is not |

20. (3 pts.) In his paper, "Design and Code Inspections to reduce errors in program development," M.E. Fagan notes that in his experience with inspections, "evidently no manager has tried to 'kill the goose that lays the golden eggs.'" Which one of the following describes the specific issue to which he was referring? (Circle ONE only.)
- a. The problem of including "bad eggs" (bad fixes or corrections) in a product due to the human tendency to consider a fix to be error-free itself.
 - b. The strong inclination of managers to change the adage that, "design is not complete until testing is completed."
 - c. Over-pressuring reviewers to seek high-occurrence, high cost error types (i.e., "golden eggs").
 - d. Pursuing questions raised during an inspection only to the point at which an error is recognized (i.e., without "killing the goose").
 - e. Zealously ensuring that all issues, problems, and concerns discovered in an inspection have been resolved by the owner.
 - f. (none of the above)

On my honor, I have neither given nor received unauthorized aid on this exam and I pledge not to divulge information regarding its contents to those who have not yet taken it.

SIGNATURE